# MSDS 6372: Project 2

## Introduction

Our goal for this project is to develop a model using logistic regression or another competing model to best predict the success of bank term deposits being sold through phone calls.

## Data Description

This project’s dataset is from UCI Machine Learning Repository from the Center for Machine Learning and Intelligent Systems. The original dataset comes from a Portugese banking institution where they utilized data mining in order to predict the success of telemarketing calls when trying to sell term deposits (Moro et. al, 2014).

The original data was collected from 2008-2013 with 150 features. This dataset has been reduced to 20 explanatory variables and one 2-level, categorical response variable, or whether the phone call resulted in a successful sell of term deposits, as provided for the purposes of this project. The explanatory variables represent four different areas of information including: bank client data, related to last contact of the current campaign, other attributes, and social and economic context attributes. There are a total of 41,188 observations with 36,548 observations with a ‘no’ response variable level and 4,640 observations with a ‘yes’ response variable level. There were no missing values to address in this dataset.

**Explanatory variable types and descriptions**

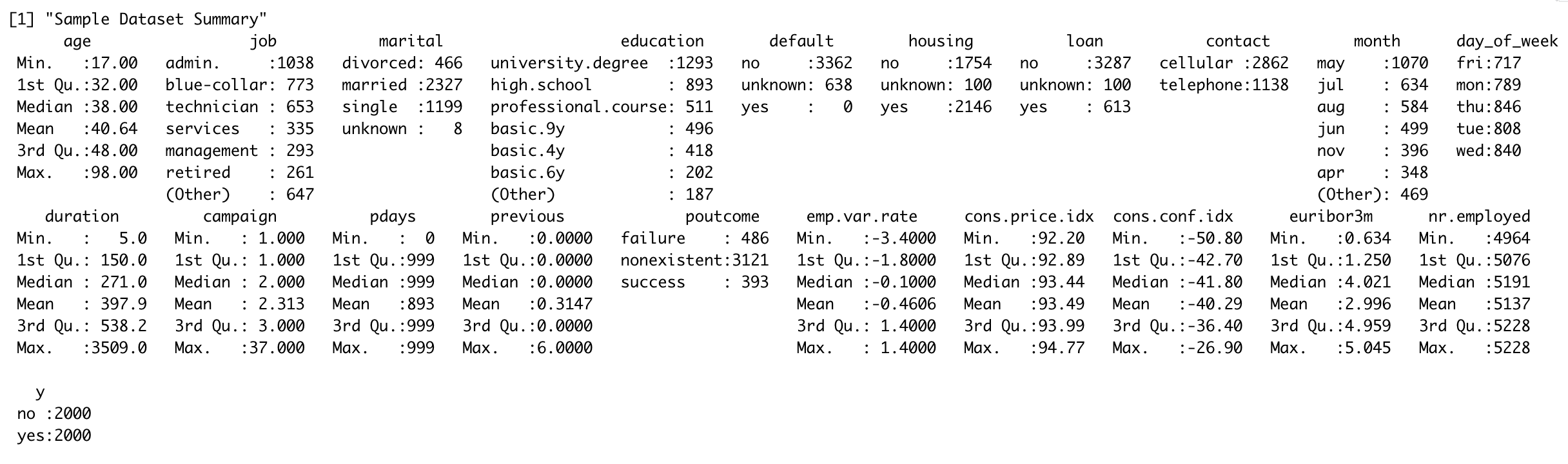
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Variable** | **Type** | **Description** |
| **Bank client data** | age | numeric | Age of client |
| job | categorical | Type of job |
| marital | categorical | marital status |
| education | categorical | Level of education |
| default | categorical | Has credit in default? |
| housing | categorical | Has housing loan? |
| loan | categorical | Has personal loan? |
| **Related to last contact of the current campaign** | contact | categorical | Contact communication type |
| month | categorical | Last contact month of year |
| day\_of\_week | categorical | last contact day of the week |
| duration | numeric | last contact duration, in seconds |
| **Other attributes** | campaign | numeric | number of contacts performed during this campaign and for this client |
| pdays | numeric | number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted) |
| previous | numeric | number of contacts performed before this campaign and for this client |
| poutcome | categorical | outcome of the previous marketing campaign |
| **Social and economic context attributes** | emp.var.rate | numeric | employment variation rate - quarterly indicator, rate at which people are being hired or fired. If economy is in recession, the number is lower because of decreasing investments. |
| cons.price.idx | numeric | consumer price index - monthly indicator, measure of the average change over time in the prices paid by urban consumers for a market basket of consumer goods. High number - inflation. Lower number - deflation. |
| cons.conf.idx | numeric | consumer confidence index - monthly indicator, interview-survey which indicates the degree of consumers’ optimism. Are they consuming or spending? Higher number - more spending. Lower number - more saving. |
| euribor3m | numeric | euribor 3 month rate - daily indicator, Euro Interbank Offered Rate rate based on averaged interest rates at which European banks borrow money from each other |
| nr.employed | numeric | number of employees - quarterly indicator |

**Down-Sampling**

To address the overwhelming number of ‘no’ responses compared to ‘yes’ responses, we decided to use down-sampling techniques. We filtered the dataset into a ‘yes’ dataset and ‘no’ dataset. We generated a random sample of 2000 observations from each dataset. We chose not to use the entire population of ‘yes’ responses and an equivalent number of ‘no’ responses in order to ensure that each sample had the same amount of randomness. We merged the 4000 observations into a single dataset for the purposes of building our models. Then split the dataset into a training and a test dataset with 3000 and 1000 observations, respectively.

## Exploratory Analysis

**Summary of Training Set**

****

**Numeric Explanatory Variables Overview**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Plots** | **Takeaways** |
| **Age** |  | Median age is roughly the same. But the yes has more right skewness than the no group |
| **Duration** |  | Yes-group has a higher median and larger distribution. Duration of the call isn’t known beforehand, therefore using it as a prediction may not be as useful. |
| **Number of contacts performed during this campaign and for this client** |  | This appears to have similar medians with the “no” group having an extended right skew tail. |
| **Days after last contact** |  | The 999 number which is used as a stand-in for no previous contact greatly skews the scale of this boxplot. |
| **Number of contacts done before** |  | The “yes” group has a larger deviation while having a similar median as the “no” group. |
| **Employment variation rate** |  | This is an interesting graph worth further exploration. Large left skewness for “no” versus the right skewness for “yes” making the medians at opposite ends. |
| **Consumer price index** |  | Medians are different with the “no” having a higher median. |
| **Consumer confidence index** |  | Similar medians. “Yes” is normally distributed with “no having right skewness. |
| **Euribor 3 Month rate** |  | Interesting plot - worth further exploration. “No” is sharply left-skewed with high median. “Yes” is sharply right skewed with a really low median. |
| **Number of employees** |  | The median and distribution of number of employees is very different. |

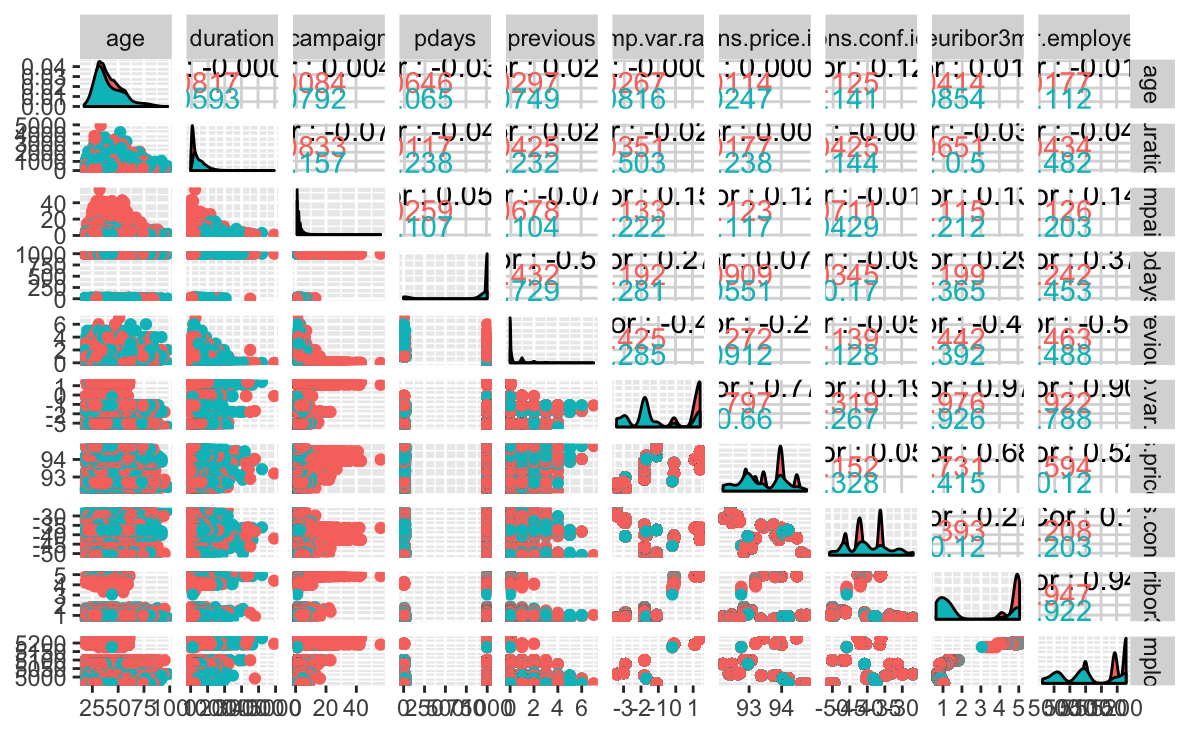
|  |  |  |
| --- | --- | --- |
| **Variable** |  | **Takeaways** |
| **Job Type** |  | Job type looks to be a possibility |
| **Marital Status** |  | Not much difference between groups. Not a variable of interest. |
| **Education Level** |  | Education level doesn’t seem to vary too much depending on success of term deposit sale. |
| **Any Default on Credit** |  | There is only one individual who has a default on credit in this sample. Recommend removing this variable because it doesn’t tell us much. |
| **Has housing loan?** |  | Not much difference between groups. Not a variable of interest. |
| **Telephone type** |  | This is interesting. there was a higher percentage of telephone users purchased term deposits compared to cell phone users. |
| **Month** |  | Timing during the years seems to be very important within this sample.. |
| **Day of week** |  | Unlike month, not much difference between days. |
| **POutcome** |  | There looks to be differences between whether or not clients will subscribe to based on previous outcomes. |

## Objective 1

### Summary of Problem and Approach

**Checking Assumptions**

It is assumed that the observations are independent. There was no evidence of multicollinearity based on the correlation plot for the continuous variables.



**Principal Component Analysis**

We attempted the principal component analysis, but did not find it useful for data reduction given that we only have 20 explanatory variables which can easily be narrowed down through other selection techniques or EDA.

### Model Selection

**Logistic Regression with Intuitive Selection**

For this model, we selected the variables based on our EDA. The suggested variables come from the blue rows in the above tables. We identified which variables were significant through running logistic regression models on individual variables and as a group. We then removed variables which were not significant according to the p-values through a process of elimination. The final model that we arrived on was  **y = contact + poutcome + emp.var.rate + cons.conf.idx + cons.price.idx**

**Parameter Interpretation**

The AIC for this model was 3314.8.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Coefficient** | **p-value** | **95% Confidence Interval** | **Interpretation** |
| (Intercept) | -106 | <.0001 | -130.794  to -83.714 | N/A |
| Contact: telephone | -.8733 | <.0001 | -1.107  to -0.6417 | When all other variables are held constant, the odds of selling a term deposit are .4175 of when the client has a landline versus a cell phone. |
| poutcome  nonexistent | 0.6184 | <.0001 | 0.3527  to 0.8861 | When all other variables are held constant, the odds of selling a term deposit increases by 1.855 when there is no previous outcome data available than an unsuccessful previous outcome. |
| poutcome  success | 1.80576 | <.0001 | 1.4019  to 2.514 | When all other variables are held constant, the odds of selling a term deposit increases by 6.896 when the contact has already purchased a term deposit than when a contact did not purchase in a previous call. |
| emp.var.rate | -.7967 | <.0001 | -0.8873  to -0.7087 | When all other variables are held constant, the odds of selling a term deposit increases by .4508 for every 1 point increase to the consumer price index. |
| cons.conf.idx | 0.06046 | <.0001 | 0.0425  to 0.0787 | When all other variables are held constant, the odds of selling a term deposit increases by 1.062 for every 1 point increase to the consumer confidence index. |
| cons.price.idx | 1.161 | <.0001 | 0.9123  to 1.419 | When all other variables are held constant, the odds of selling a term deposit increases by 3.193 for every 1 point increase to the consumer price index. |

**Final Conclusions**

If a bank is interested in being strategic about how to utilize their staff by prioritizing possible clients to purchase term deposits, the bank should consider maintaining previous clients who have purchased term deposits, since the odds (6.896) are they are most likely out of all variables to purchase a term deposit. Another interesting factor was whether the client had a telephone or cell phone. If the client had a telephone they were more likely to purchase a term deposit with odds increase of .4175. Other variables of interest are the relationship of timing with larger external factors such as the employee variation rate, consumer price index, and consumer confidence index. The consumer price index could influence a client’s success by 3.193 times more over the odds of a 1-point lower consumer price index time period.

## Objective 2

**Overall Approach**

We built a series of models to compare against the intuitive model and each other. The models included were adding complexity to our intuitive model through an interaction term, logistic regression with stepwise, backward and forward selection, linear discriminant analysis, random forest, and knn. We then used the same test set for measuring the prediction accuracy and generated the confusion matrix for each.

**Logistic Regression with Complexity**

We created a complex model which utilized an interaction term. The variables were selected based on EDA and proven significance in other models.

Final Model:

y = euribor3m + cons.price.idx + poutcome + cons.conf.idx + cons.price.idx\*cons.conf.idx

**Selection Models**

In the following models, we used stepwise, forward, and backwards selection to determine the best fitting model based off of AIC or VIF.

***Stepwise***

Final model:

y = age + default + contact + month + day\_of\_week + duration + campaign + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + nr.employed

AIC: 3240.412

***Forwards***

Final model:

y ~ age + job + marital + education + default + housing + loan + contact + month + day\_of\_week + campaign + pdays + previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed

AIC: 3276.5

***Backwards***

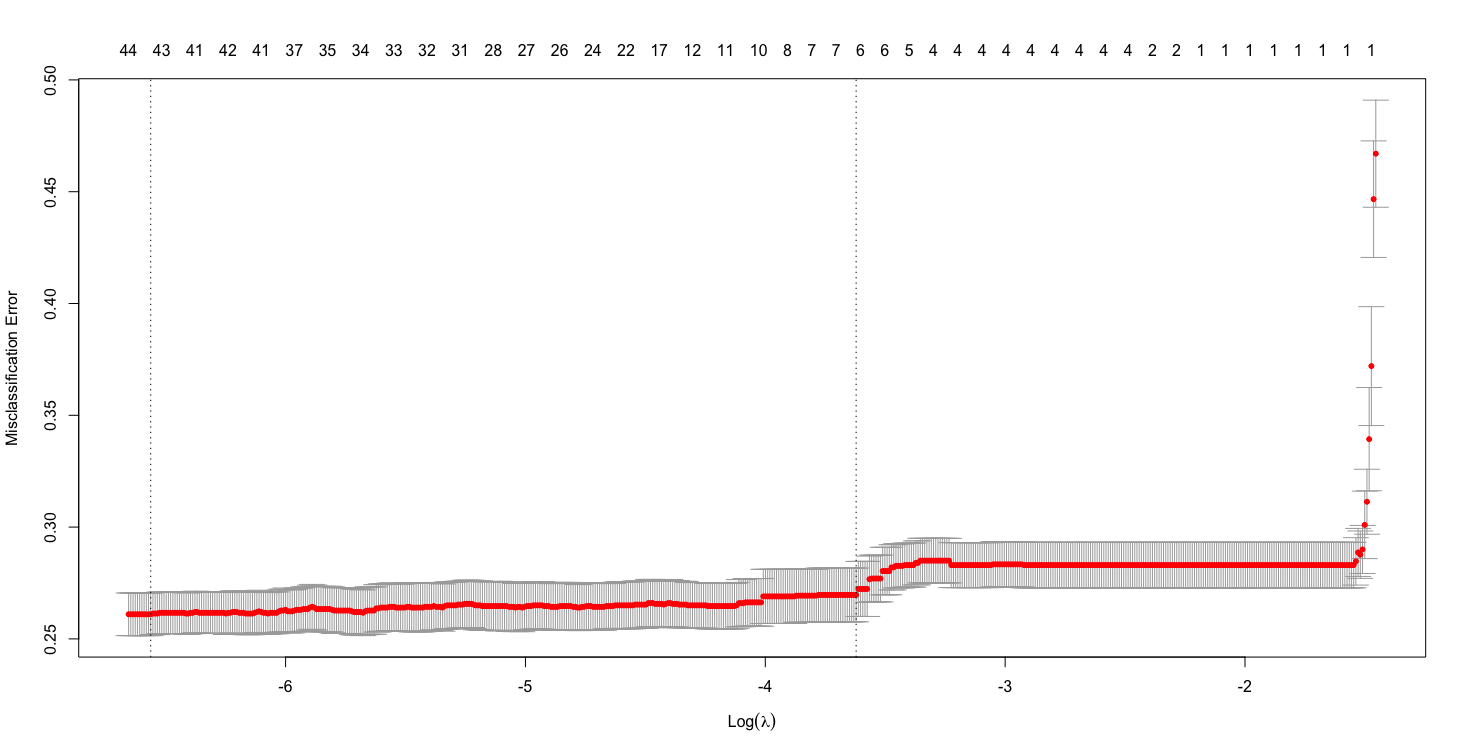
Final model:

y ~ housing + contact + month + day\_of\_week + pdays + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + nr.employed

AIC: 3240.4

**LASSO Model**

We have used a lambda sequence of 1000 and we got 0.262 as CV error rate while using LASSO.

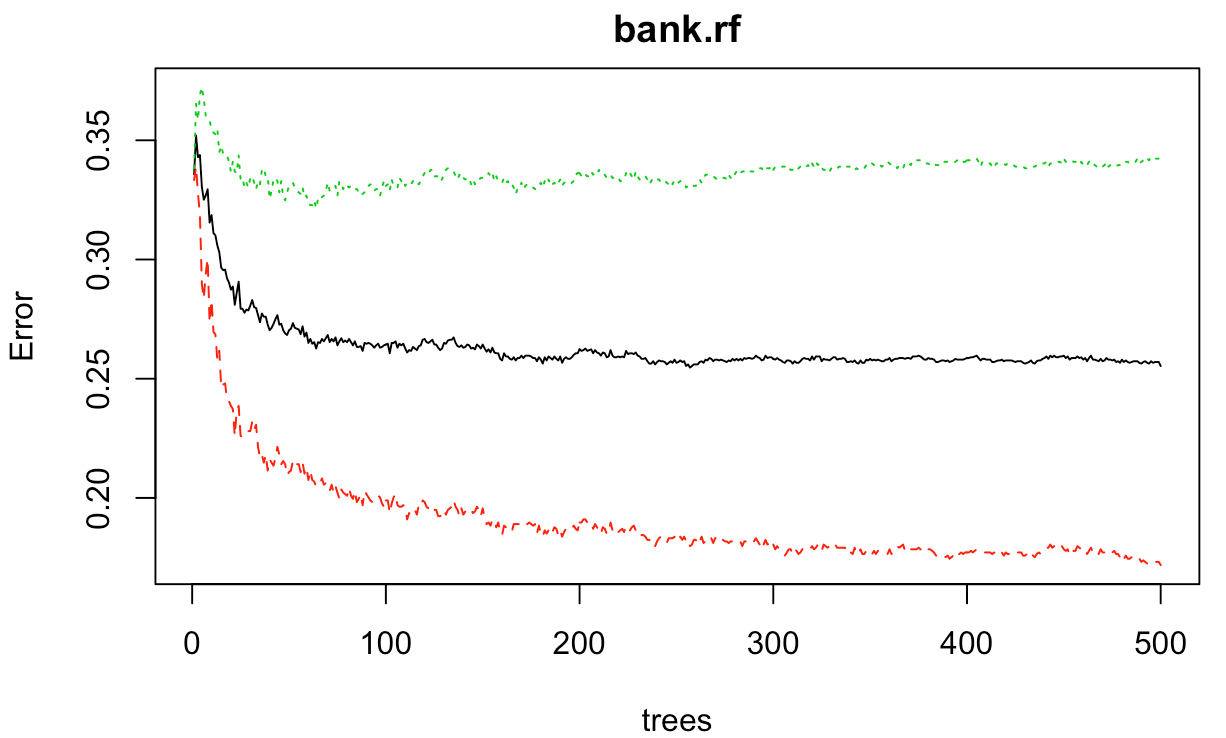
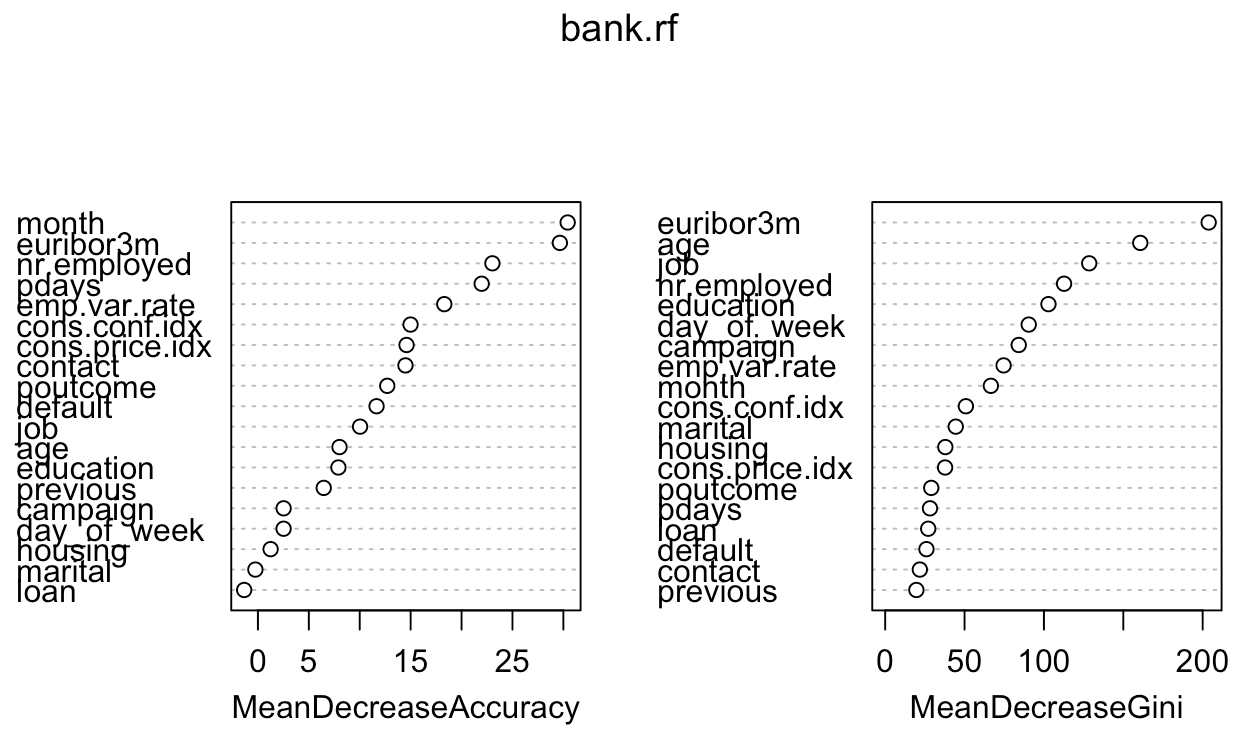


**LDA Model**

We have decided that in the interest of the bank, utilizing more staff hours to reach out to clients who we anticipate purchasing would be more cost-effective than losing money by not calling clients that will potentially purchase term deposits. Therefore, we adjusted our sensitivity so that we would get more false positives.

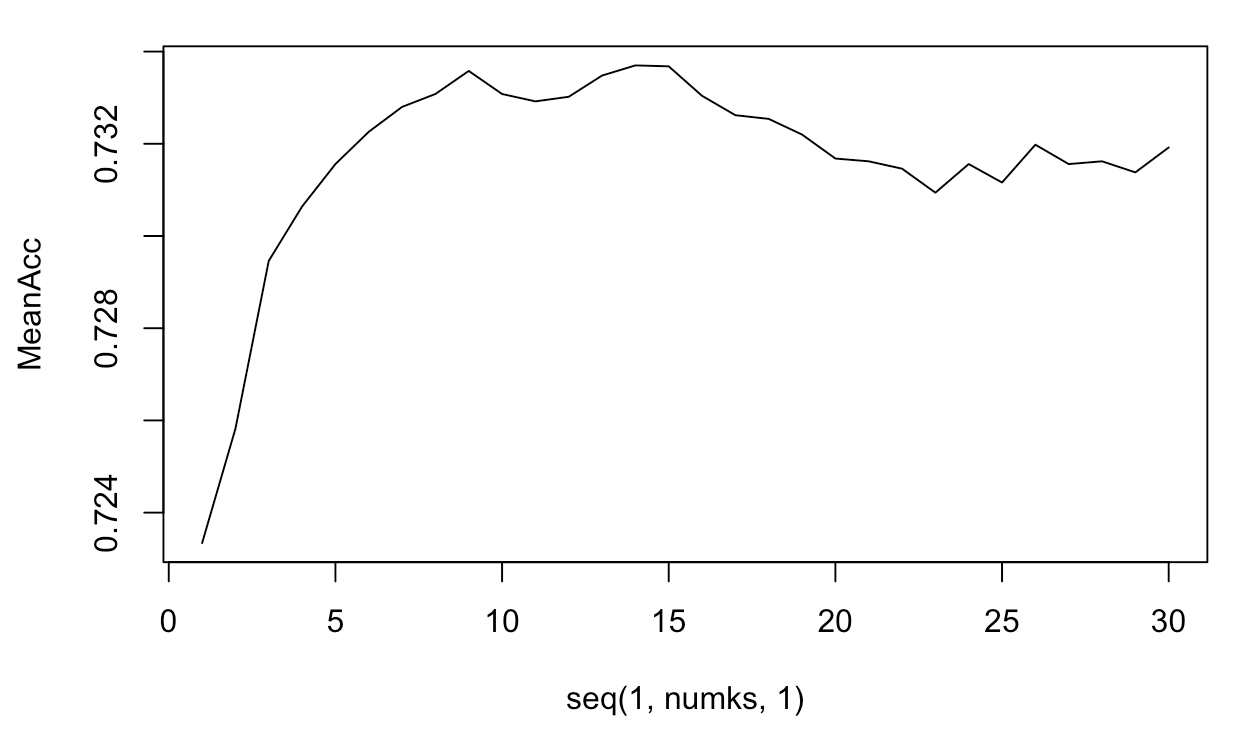
**Random Forest**

For the random forest, all variables were considered except for duration. The following are the tables for the random forest that depict the error rates and mean accuracy based on the variables.

**knn Model**

We built the knn model using the continuous variables cons.price.idx, cons.conf.idx, and euribor3m, since they were consistently promising from our EDA and intuitive logistic regression model. After performing hypertuning on the knn model for k=1 through k=30, we determined that k=10 would be the most optimal level for this model based on the following charts.



### Main Analysis

The model performance chart below provides the comparison of the models selected as candidates for the final model. The chart maps out the sensitivity, specificity, accuracy, error rate, and McNemar’s p-value for each of the models. The blue cells highlight the best value for each column, while the red indicates the worst value in each column.

**Model Performance Chart**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Sensitivity** | **Specificity** | **Overall Accuracy** | **Error Rate** | **McNemar’s** | **Confusion Matrix Output** |
| Intuitive/  Simple | .6238 | .7906 | .732 | .295 | 1.606 e-07 |  |
| Complex/ Interaction Term | .6160 | .7741 | .693 | .307 | 9.188e-7 |  |
| Stepwise | .5945 | .8439 | .716 | .289 | 7.639e-15 |  |
| LASSO | .6023 | .8337 | .715 | .285 | 4.95e-13 |  |
| LDA | .7895 | .8296 | .809 | .191 | .08246 |  |
| Random Forest | .6121 | .8398 | .723 | .277 | 5.59e-13 |  |
| knn(k=10) | .8022 | .6490 | .704 | .296 | <2.2e-16 |  |

## Conclusion/Discussion

The question of interest for this project was to understand what factors were useful to consider when determining in advance the success of a telemarketing call in subscribing a client to a term deposit purchase.

For the purpose of this project, we opted for models with less sensitivity and greater specificity because we were interested in having greater false positives than false negatives given the nature of the sales. Our decision was based on having a greater number of false positives since we were willing to balance the extra staff time making telemarketing calls rather than risk losing potential clients.

We selected the LDA model as our final model. The overall accuracy (80.9%) and error rate (19.1%) were the best out of all models. While the McNemar’s p-value was not significant, we did not find this metric useful because the LDA model outperformed the other models. Even considering McNemar’s, the .08 value was close to our .05 threshold. Furthermore, the balance of specificity and sensitivity matched our desired practical needs.

Future analysis and potential opportunities for further exploration would be to meet with the collectors of the data to understand the way that some of these variables were collected. Some of the variables that were considered continuous could in fact be categorical, such as the campaign. Some of the continuous variables could also be converted into categorical through defining high or low variables, such as the Euribor 3month Average Interest Rates and then we could make more use of knn or random forest if we already have some predefined groups that can assist.

## Appendix

### Bibliography

S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

UCI Machine Learning Repository. Bank Marketing Data Set. <https://archive.ics.uci.edu/ml/datasets/bank%20marketing>. Date Access: August 16, 2020.

### R code

# Bank Marketing Data Set

```{r setup, include=FALSE}

#knitr::opts\_chunk$set(warning = FALSE)

knitr::opts\_chunk$set(echo = FALSE, message = FALSE, warning = FALSE)

library(gplots)

library(ggplot2)

library(tidyverse)

library(naniar)

library(plyr)

library(readr)

library(dplyr)

library(MASS)

library(GGally)

library(randomForest)

library(lda)

library(e1071)

library(caret)

library(class)

library(ROCR)

library(car)

library(glmnet)

```

```{r Reading datafile}

# Reading datafile

bank.additional.full <- read\_delim("../Data/bank-additional-full.csv", delim = ";")

```

```{r}

#Missing data?

vis\_miss(bank.additional.full)

```

```{r dataset setup}

#### To ensure performance metrics are comparable across models, setting the training and test sets here in the beginning.

#Converting the categorical variables as factors

bank.additional.full$y <- as.factor(bank.additional.full$y)

bank.additional.full$job <- as.factor(bank.additional.full$job)

bank.additional.full$marital <- as.factor(bank.additional.full$marital)

bank.additional.full$education <- as.factor(bank.additional.full$education)

bank.additional.full$default <- as.factor(bank.additional.full$default)

bank.additional.full$housing <- as.factor(bank.additional.full$housing)

bank.additional.full$loan <- as.factor(bank.additional.full$loan)

bank.additional.full$contact <- as.factor(bank.additional.full$contact)

bank.additional.full$day\_of\_week <- as.factor(bank.additional.full$day\_of\_week)

bank.additional.full$poutcome <- as.factor(bank.additional.full$poutcome)

bank.additional.full$month <- as.factor(bank.additional.full$month)

#Removing records from bank.additional.full where value of default = 'yes' as it might cause issue when picked in non-balanced way.

bank.additional.full <- bank.additional.full %>% filter (default!='yes')

# Separating out yes and no observations

bank.additional.no <- bank.additional.full %>% filter (y=='no')

bank.additional.yes <- bank.additional.full %>% filter (y=='yes')

# Picking 1000 sample each from yes and no

set.seed(1234)

index.no<-sample(1:nrow(bank.additional.no),2000,replace=FALSE)

index.yes<-sample(1:nrow(bank.additional.yes),2000,replace=FALSE)

bank.additional.sample.no<-bank.additional.no[index.no,]

bank.additional.sample.yes<-bank.additional.yes[index.yes,]

bank.additional.sample <- rbind(bank.additional.sample.no,bank.additional.sample.yes)

#Splitting train and test data set

set.seed(1234)

index<-sample(1:4000,3000,replace=FALSE)

bank.additional.sample <- as.data.frame(bank.additional.sample)

bank.additional.sample.train<-bank.additional.sample[index,]

bank.additional.sample.train <- as.data.frame(bank.additional.sample.train)

bank.additional.sample.test <-bank.additional.sample[-index,]

bank.additional.sample.test <- as.data.frame(bank.additional.sample.test)

```

```{r summary of dataset, include=FALSE}

#summary of dataset

print("Full Dataset Summary")

summary(bank.additional.full)

print("Sample Dataset Summary")

summary(bank.additional.sample)

print("Training Dataset Summary")

summary(bank.additional.sample.train)

```

```{r visualizing relationship between y and continous variables}

#boxplot of response variable (y) versus all other continous variables

bank.additional.sample %>% ggplot(aes(x = y, y=age)) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success by Age") + ylab("Age")

bank.additional.sample %>% ggplot(aes(x = y, y=duration)) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success by duration") + ylab("duration")

bank.additional.sample %>% ggplot(aes(x = y, y=campaign)) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success by campaign") + ylab("campaign")

bank.additional.sample %>% ggplot(aes(x = y, y=pdays)) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success") + ylab("Days after last contact")

bank.additional.sample %>% ggplot(aes(x = y, y=previous)) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success ") + ylab("Number of contacts done before")

bank.additional.sample %>% ggplot(aes(x = y, y=emp.var.rate )) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success ") + ylab("employment variation rate - quarterly indicator")

bank.additional.sample %>% ggplot(aes(x = y, y=cons.price.idx)) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success ") + ylab("consumer price index - monthly indicator")

bank.additional.sample %>% ggplot(aes(x = y, y=cons.conf.idx)) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success ") + ylab("consumer confidence index - monthly indicator")

bank.additional.sample %>% ggplot(aes(x = y, y=euribor3m)) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success") + ylab("euribor 3 month rate - daily indicator")

bank.additional.sample %>% ggplot(aes(x = y, y=nr.employed)) + geom\_boxplot(fill="red") + labs(title = "Term Deposits Success ") + ylab("number of employees - quarterly indicator")

```

```{r ftable categorical predictors, include=FALSE}

#Visualize Categorical variables

#Table of counts are helpful for categorical variables

attach(bank.additional.sample)

ftable(addmargins(table(y,job)))

ftable(addmargins(table(y,marital)))

ftable(addmargins(table(y,education)))

ftable(addmargins(table(y,default)))

ftable(addmargins(table(y,housing)))

ftable(addmargins(table(y,loan)))

ftable(addmargins(table(y,contact)))

ftable(addmargins(table(y,month)))

ftable(addmargins(table(y,day\_of\_week)))

ftable(addmargins(table(y,poutcome)))

```

```{r visualizing categorical predictors}

#to get proportions that make sense

prop.table(table(y,job),2)

plot(y~job,col=c("red","blue"))

prop.table(table(y,marital),2)

plot(y~marital,col=c("red","blue"))

prop.table(table(y,education),2)

plot(y~education,col=c("red","blue"))

prop.table(table(y,default),2)

plot(y~default,col=c("red","blue"))

prop.table(table(y,housing),2)

plot(y~housing,col=c("red","blue"))

prop.table(table(y,loan),2)

plot(y~loan,col=c("red","blue"))

prop.table(table(y,contact),2)

plot(y~contact,col=c("red","blue"))

prop.table(table(y,month),2)

plot(y~month,col=c("red","blue"))

prop.table(table(y,day\_of\_week),2)

plot(y~day\_of\_week,col=c("red","blue"))

prop.table(table(y,poutcome),2)

plot(y~poutcome,col=c("red","blue"))

```

#####Logistics Regression Assumption Check

```{r checking assumption before fitting logistic regression}

#Logistics Regression Assumption Check

#Overall EDA to check collinearity between continous variables

ggpairs(bank.additional.full,columns=c(1,11:14,16:20),aes(colour=y))

```

#### Perform your logistic regression analysis and provide interpretation of the regression coefficients including hypothesis testing, and confidence intervals.

```{r logistic regression on individual predictors, include=FALSE}

# Checking significance of individual variables in predicting outcome (Suchi)

summary(glm(y~age , family="binomial",data=bank.additional.sample.train)) ##From boxplot, it did not look significant but from glm, it does look OK.

summary(glm(y~job , family="binomial",data=bank.additional.sample.train))

summary(glm(y~marital , family="binomial",data=bank.additional.sample.train)) ## Not at all

summary(glm(y~education , family="binomial",data=bank.additional.sample.train)) ## Meh

summary(glm(y~default , family="binomial",data=bank.additional.sample.train)) ## Okay, will get back to you!!!!

summary(glm(y~housing , family="binomial",data=bank.additional.sample.train)) ## Not at all

summary(glm(y~loan , family="binomial",data=bank.additional.sample.train)) ## Not at all

summary(glm(y~contact , family="binomial",data=bank.additional.sample.train))

summary(glm(y~month , family="binomial",data=bank.additional.sample.train))

summary(glm(y~day\_of\_week , family="binomial",data=bank.additional.sample.train))## Not at all

summary(glm(y~duration , family="binomial",data=bank.additional.sample.train)) ## This is not a true predictor

summary(glm(y~campaign , family="binomial",data=bank.additional.sample.train))

summary(glm(y~pdays , family="binomial",data=bank.additional.sample.train))

summary(glm(y~previous , family="binomial",data=bank.additional.sample.train))

summary(glm(y~poutcome , family="binomial",data=bank.additional.sample.train))

summary(glm(y~emp.var.rate , family="binomial",data=bank.additional.sample.train))

summary(glm(y~cons.price.idx , family="binomial",data=bank.additional.sample.train))

summary(glm(y~cons.conf.idx , family="binomial",data=bank.additional.sample.train))

summary(glm(y~euribor3m , family="binomial",data=bank.additional.sample.train))

summary(glm(y~nr.employed , family="binomial",data=bank.additional.sample.train))

```

`Type of Selection : (Manual / Intuition)`

```{r}

#Type of Selection (Manual / Intuition)

#fitting a simple model by using only those predictors which appeared to be significant in EDA and individual variable check -- ##Kind of Manual backward selection

simple.logistics.all.significant<-glm(y~ job + contact + month + campaign + pdays + previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed, family="binomial",data=bank.additional.sample.train)

summary(simple.logistics.all.significant)

#confint(simple.logistics.all.significant)

#Removing job variable as it does not look statistically significant in model (Got lowest AIC 3270 here)

simple.logistics.all1 <-glm(y~ contact + month + campaign + pdays + previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed, family="binomial",data=bank.additional.sample.train)

summary(simple.logistics.all1)

#confint(simple.logistics.all1)

#Removing month variable as it does not look practically significant

simple.logistics.all2 <-glm(y~ contact + campaign + pdays + previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed, family="binomial",data=bank.additional.sample.train)

summary(simple.logistics.all2)

#confint(simple.logistics.all2)

#Removing campaign variable as it does not look statistically significant in model

simple.logistics.all3 <-glm(y~ contact + pdays + previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed, family="binomial",data=bank.additional.sample.train)

summary(simple.logistics.all3)

#confint(simple.logistics.all3)

#Removing pdays variable as it does not look statistically significant in model

simple.logistics.all4 <-glm(y~ contact + previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed, family="binomial",data=bank.additional.sample.train)

summary(simple.logistics.all4)

#confint(simple.logistics.all4)

#Removing previous variable as it does not look statistically significant in model

simple.logistics.all5 <-glm(y~ contact + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed, family="binomial",data=bank.additional.sample.train)

summary(simple.logistics.all5)

#confint(simple.logistics.all5)

#Removing nr.employed variable as it does not look statistically significant in model and it looked like related to emp.var.rate

simple.logistics.all6 <-glm(y~ contact + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m, family="binomial",data=bank.additional.sample.train)

summary(simple.logistics.all6)

#confint(simple.logistics.all6)

#Removing euribor3m variable as it does not look statistically significant in model and it looked like related to emp.var.rate

simple.logistics<-glm(y~ contact + poutcome + emp.var.rate + cons.conf.idx + cons.price.idx , family="binomial",data=bank.additional.sample.train)

summary(simple.logistics)

confint(simple.logistics)

```

Since our goal is to predict the outcome, removing the `duration` variable from stepwise full model even though it looked significant in the full model.

Practically I do not believe customer response depends on which month or day\_of\_week they were contacted (could be just a coincidence). But for now keeping it in the model.

Finalizing the below equation as it had lowest AIC among Forward, Backward and Stepwise model. Backward and Stepwise has exactly same set of predictor variables.

`y ~ housing + contact + month + day\_of\_week + pdays + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + nr.employed`

```{r}

#Type of Selection : Stepwise, Forward, Backward

# First Fit the full model (Excluding duration variable)

full.log.model<-glm(y~.,family="binomial",data=bank.additional.sample.train[-11])

# Stepwise model

print("Stepwise Model Details:")

step.model <- stepAIC(full.log.model, direction = "both", trace = FALSE)

summary(step.model)

print("Step Model AIC:" )

AIC(step.model)

exp(cbind("Odds ratio" = coef(step.model), confint.default(step.model, level = 0.95)))

vif(step.model)

# forward regression model

print("Forward Model Details:")

foward.model <- stepAIC(full.log.model, direction = "forward", trace = FALSE)

print("Forward Model AIC:" )

AIC(foward.model)

summary(foward.model)

# backward regression model

print("Backward Model Details:")

backward.model <- stepAIC(full.log.model, direction = "backward", trace = FALSE)

print("Backward Model AIC:" )

AIC(backward.model)

summary(backward.model)

```

```{r include=FALSE}

#fitting a simple model by using only 4 predictors which appeared to be significant in EDA and p-values from previous step (Hollie)

simple.logistics.trained<-glm(y~ euribor3m + cons.price.idx + poutcome + cons.conf.idx, family="binomial",data=bank.additional.sample.train)

summary(simple.logistics.trained)

confint(simple.logistics.trained)

```

```{r}

# Performance comparison for simple logistic model

#simple.logistics.trained<-glm(y~ euribor3m + cons.price.idx + poutcome + cons.conf.idx, family="binomial",data=bank.additional.sample.train)

fit.simple.logistics.pred<-predict(step.model, newdata = bank.additional.sample.test, type = "response")

cutoff<-0.5

class.simple.logistics<-factor(ifelse(fit.simple.logistics.pred>cutoff,"yes","no"),levels=c("no","yes"))

conf.simple.logistics<-confusionMatrix(table(class.simple.logistics,bank.additional.sample.test$y), positive = 'yes')

conf.simple.logistics

results.simple.logistics<-prediction(fit.simple.logistics.pred,bank.additional.sample.test$y,label.ordering=c("no","yes"))

roc.simple.logistics <- performance(results.simple.logistics, measure = "tpr", x.measure = "fpr")

plot(roc.simple.logistics,colorize = TRUE)

abline(a=0, b=1)

```

Removing duration variable here before trying LASSO to compare the performance with stepwise

```{r}

# LASSO

dat.train.x <- model.matrix(y ~ .,bank.additional.sample.train[-11])

dat.train.y<-bank.additional.sample.train[,21]

cvfit <- cv.glmnet(dat.train.x, dat.train.y, family = "binomial", type.measure = "class", nlambda = 1000)

plot(cvfit)

coef(cvfit, s = "lambda.min")

print("CV Error Rate:")

cvfit$cvm[which(cvfit$lambda==cvfit$lambda.min)]

#Optimal penalty

print("Penalty Value:")

cvfit$lambda.min

#For final model predictions go ahead and refit lasso using entire

#data set

lasso.log.final<-glmnet(dat.train.x, dat.train.y, family = "binomial",lambda=cvfit$lambda.min)

dat.test.x<-model.matrix(y ~ .,bank.additional.sample.test[-11])

fit.pred.lasso <- predict(lasso.log.final, newx = dat.test.x, type = "response")

fit.pred.step<-predict(step.model,newdata=bank.additional.sample.test,type="response")

```

`Confusion matrix for LASSO`

```{r}

#confusion matrix for LASSO

#Using cutoff of 0.5 to make the classification.

cutoff<-0.5

class.lasso<-factor(ifelse(fit.pred.lasso>cutoff,"yes","no"),levels=c("no","yes"))

class.step<-factor(ifelse(fit.pred.step>cutoff,"yes","no"),levels=c("no","yes"))

#Confusion Matrix for Lasso

conf.lasso<-table(class.lasso,bank.additional.sample.test$y)

print("Confusion matrix for LASSO")

conf.lasso

conf.step<-table(class.step,bank.additional.sample.test$y)

print("Confusion matrix for Stepwise")

conf.step

```

```{r}

#Overall Accuracy of LASSO and Stepwise

print("Overall accuracy for LASSO and Stepwise respectively")

sum(diag(conf.lasso))/sum(conf.lasso)

sum(diag(conf.step))/sum(conf.step)

```

```{r}

# ROC plot for LASSO

results.lasso<-prediction(fit.pred.lasso, bank.additional.sample.test$y,label.ordering=c("no","yes"))

roc.lasso <- performance(results.lasso, measure = "tpr", x.measure = "fpr")

plot(roc.lasso,colorize = TRUE)

abline(a=0, b= 1)

```

### Objective 2: You must include one additional logistic regression model which is also a more complicated logistic regression model than in Objective 1.

`Fitting a complex model using 4 variables and 1 interaction`

```{r}

#Fitting a complex model using 4 variables and 1 interaction (Kenny)

complex.interaction<-glm(y~ euribor3m + cons.price.idx + poutcome + cons.conf.idx + cons.price.idx\*cons.conf.idx, family="binomial",data=bank.additional.sample.train)

summary(complex.interaction)

confint(complex.interaction)

```

```{r}

# Performance comparison for simple logistic model + interaction term

complex.interaction.trained<-glm(y~ euribor3m + cons.price.idx + poutcome + cons.conf.idx + cons.price.idx\*cons.conf.idx, family="binomial",data=bank.additional.sample.train)

fit.complex.interaction.pred<-predict(complex.interaction.trained, newdata = bank.additional.sample.test, type = "response")

cutoff<-0.5

class.complex.interaction<-factor(ifelse(fit.complex.interaction.pred>cutoff,"yes","no"),levels=c("no","yes"))

# conf.complex.interaction<-table(class.complex.interaction,bank.additional.sample.test$y)

conf.complex.interaction<-confusionMatrix(table(class.complex.interaction,bank.additional.sample.test$y), positive = 'yes')

conf.complex.interaction

# mean(class.complex.interaction==bank.additional.sample.test$y)

results.complex.interaction<-prediction(fit.complex.interaction.pred,bank.additional.sample.test$y,label.ordering=c("no","yes"))

roc.complex.interaction = performance(results.complex.interaction, measure = "tpr", x.measure = "fpr")

plot(roc.complex.interaction,col = "blue")

plot(roc.simple.logistics,col = "red",add=TRUE)

plot(roc.lasso,col = "green",add=TRUE)

legend("bottomright",legend=c("Simple","Added Interaction", "LASSO"),col=c("blue","red","green"),lty=1,lwd=1)

abline(a=0, b=1)

```

#### Create another competing model using just the continuous predictors and use LDA or QDA.

`LDA`

```{r}

#Objective 2 (Point 3) -- LDA on Sample data (Objective 2: Point 3) # Did not use duration variable

bank.additional.lda <- lda(y ~ ., bank.additional.sample.train[c(1,10:14,16:21)])

bank.additional.lda.p <- predict(bank.additional.lda, bank.additional.sample.test)$class

table.lda <- table(bank.additional.lda.p, bank.additional.sample.test$y)

cm.lda = confusionMatrix(table.lda, positive = 'yes')

cm.lda

```

#### (Optional) Use a nonparameteric model approach as a competing model. Random forest or decision tree for predictors that are both categorical and continuous.

`Random Forest`

```{r}

#Random Forest (Suchi: Optional) # Did not use duration variable

bank.rf<- randomForest(y~., data = bank.additional.sample.train[-11], importance=TRUE)

bank.rf

varImpPlot(bank.rf)

plot(bank.rf)

bank.rf.pred<- predict(bank.rf, bank.additional.sample.test)

##Confusion Matrix

table.rf <- table(bank.rf.pred,bank.additional.sample.test$y)

cm.rf = confusionMatrix(table.rf, positive = 'yes')

cm.rf

```

#### (Optional) Use a nonparameteric model approach as a competing model. k-nearest neighbors approach if just working with continuous predictors.

After running thru 50 iterations on same sample data set and running thru 100 iteration on different split dataset, we see that at K=10, the model has highest performance.

```{r knn}

#KNN

set.seed(123)

splitPerc = .75

iterations = 100

numks = 50

masterAcc = matrix(nrow = iterations, ncol = numks)

for(j in 1:iterations)

{

accs = data.frame(accuracy = numeric(50), k = numeric(50))

trainIndices = sample(1:dim(bank.additional.sample)[1],round(splitPerc \* dim(bank.additional.sample)[1]))

train = bank.additional.sample[trainIndices,]

test = bank.additional.sample[-trainIndices,]

for(i in 1:numks)

{

classifications = knn(train[,c(17,19)],test[,c(17,19)],train$y, prob = TRUE, k = i)

table(classifications,test$y)

CM = confusionMatrix(table(classifications,test$y))

masterAcc[j,i] = CM$overall[1]

}

}

MeanAcc = colMeans(masterAcc)

plot(seq(1,numks,1),MeanAcc, type = "l")

## Loop for many k and one training / test partition

accs = data.frame(accuracy = numeric(50), k = numeric(50))

for(i in 1:50)

{

classifications = knn(bank.additional.sample.train[,c(17,19)],bank.additional.sample.test[,c(17,19)],bank.additional.sample.train$y, prob = TRUE, k = i)

table(bank.additional.sample.test$y,classifications)

CM = confusionMatrix(table(bank.additional.sample.test$y,classifications),positive = 'yes')

accs$accuracy[i] = CM$overall[1]

accs$k[i] = i

}

plot(accs$k,accs$accuracy, type = "l", xlab = "k")

# k = 10 (optimal that we found)

classifications = knn(bank.additional.sample.train[,c(17,20)],bank.additional.sample.test[,c(17,20)],bank.additional.sample.train$y, prob = TRUE, k = 10)

table(bank.additional.sample.test$y,classifications)

confusionMatrix(table(bank.additional.sample.test$y,classifications) ,positive = 'yes')

cm.knn = confusionMatrix(table(bank.additional.sample.test$y,classifications),positive = 'yes')

cm.knn$overall[1]

```

#### Provide a summary table of the performance across the competing methods. Summarize the overall findings.

```{r }

#Comparing metrics from all model

print("Simple model Overall Accuracy:")

conf.simple.logistics$overall[1]

print("Complex model Overall Accuracy:")

conf.complex.interaction$overall[1]

print("Random Forest Overall Accuracy:")

cm.rf$overall[1]

print("LDA Overall Accuracy:")

cm.lda$overall[1]

print("KNN Overall Accuracy:")

cm.knn$overall[1]

print("Simple model Misclassification rate:")

1-sum(diag(conf.simple.logistics$table))/sum(conf.simple.logistics$table)

print("Complex model Misclassification rate:")

1-sum(diag(conf.complex.interaction$table))/sum(conf.complex.interaction$table)

print("RandomForest Misclassification rate:")

1-sum(diag(cm.rf$table))/sum(cm.rf$table)

print("LDA Misclassification rate:")

1-sum(diag(cm.lda$table))/sum(cm.lda$table)

print("KNN Misclassification rate:")

1-sum(diag(cm.knn$table))/sum(cm.knn$table)

```

The Rmd can be found in <https://github.com/msuchismita/DS_6372_Project2/blob/master/R-Markdown/6372-project-2.Rmd>